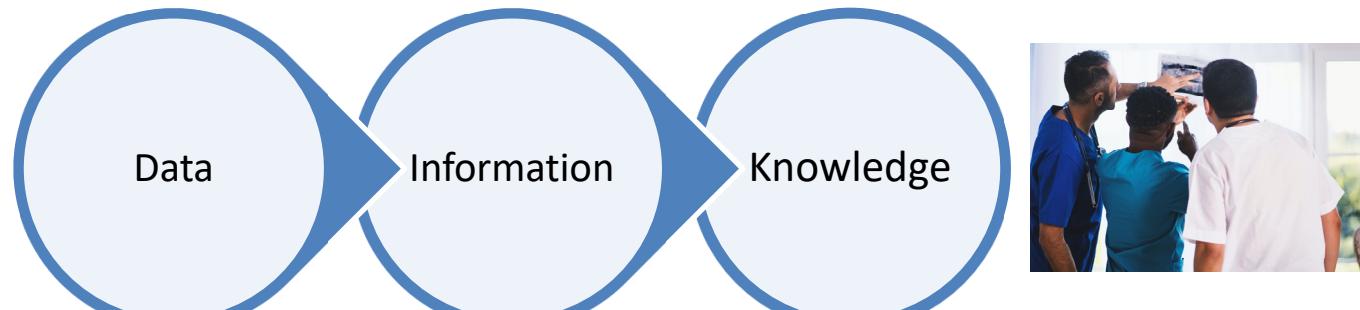


Introduction to Data Science: Data and Challenges in Managing Data



Introduction to Data Science: Data Transformation



Introduction to Data Science: Data Sources

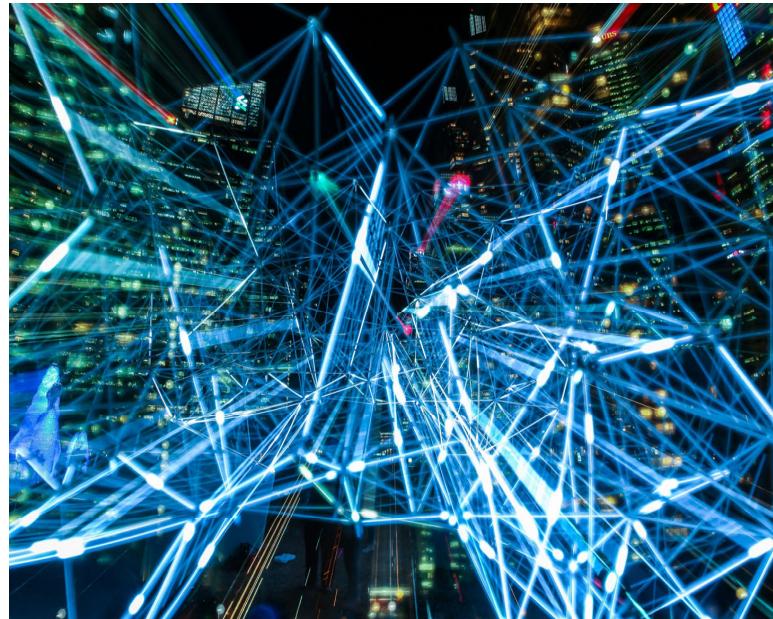


Introduction to Data Science: Big Data

“At the beginning of **2020**, the digital universe was estimated to consist of 44 zettabytes of **data**. By 2025, approximately 463 exabytes would be **created every** 24 hours worldwide.”

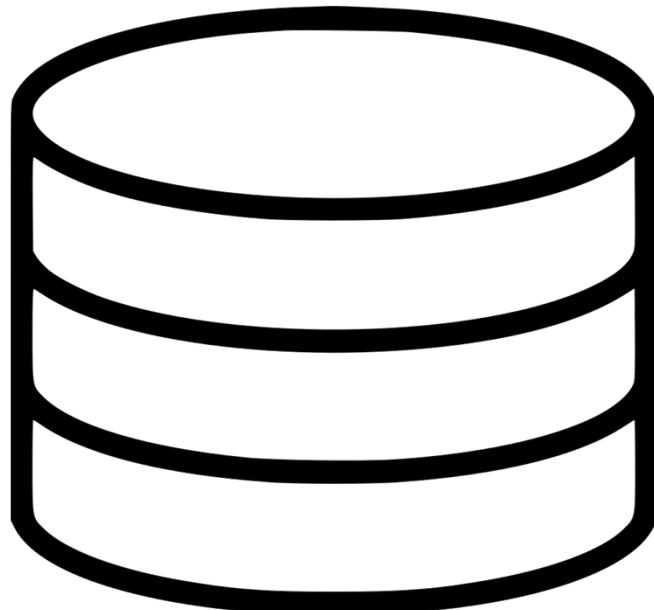
Source: seedscientific.com

Introduction to Data Science: Challenges in Managing Data



- Amount of data increases exponentially
- Data are scattered, unstructured, and from multiple sources
- Data security, quality and integrity issues

Introduction to Data Science: Challenges in Managing Data

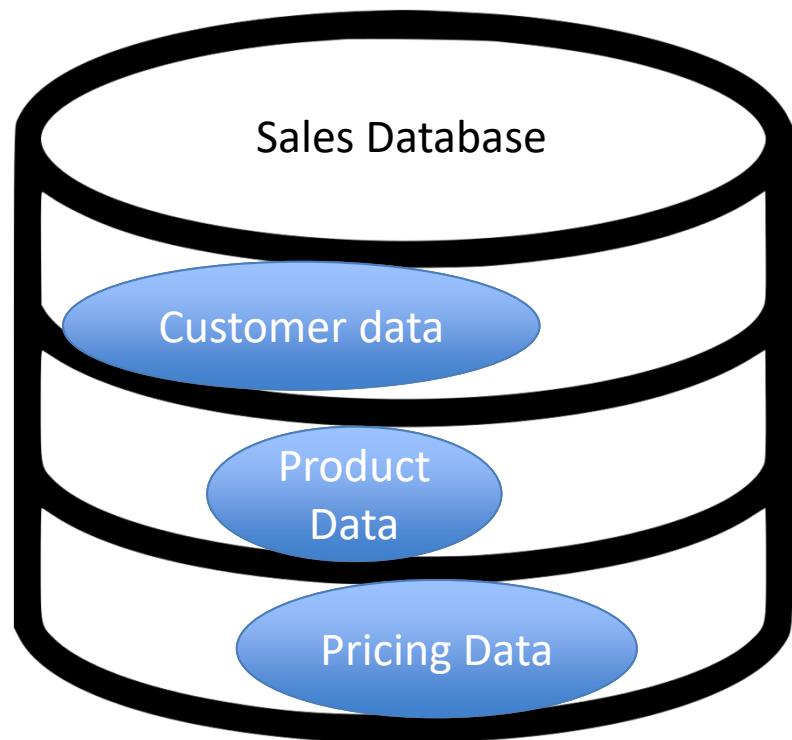


Concept of Database to organize and
manage Data

Introduction to Data Science: Database Concept

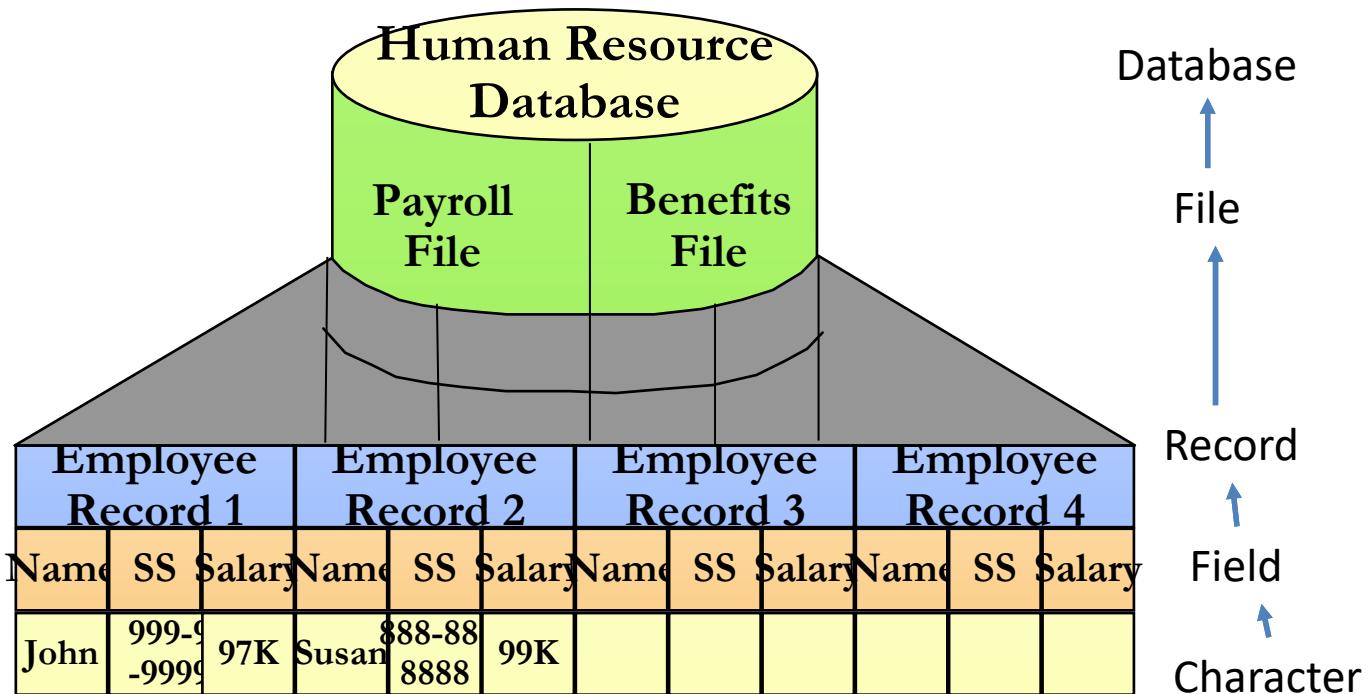


Introduction to Data Science: Database Concept

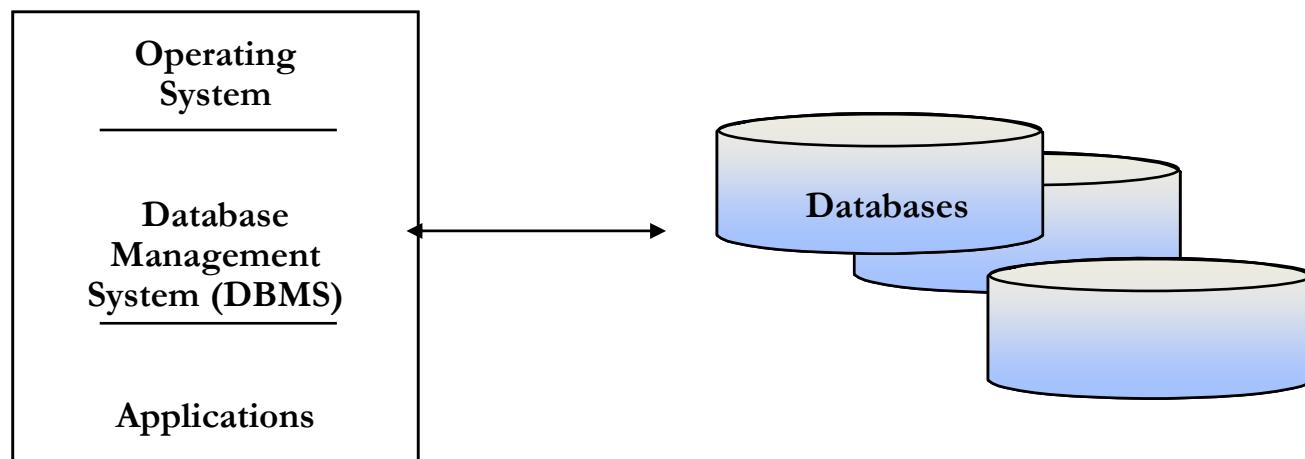


A database is an **integrated collection** of logically related records or objects.

Introduction to Data Science: Relational Database



Introduction to Data Science: Database Concept



Introduction to Data Science: Database Management System (DBMS)

Functions of a DBMS

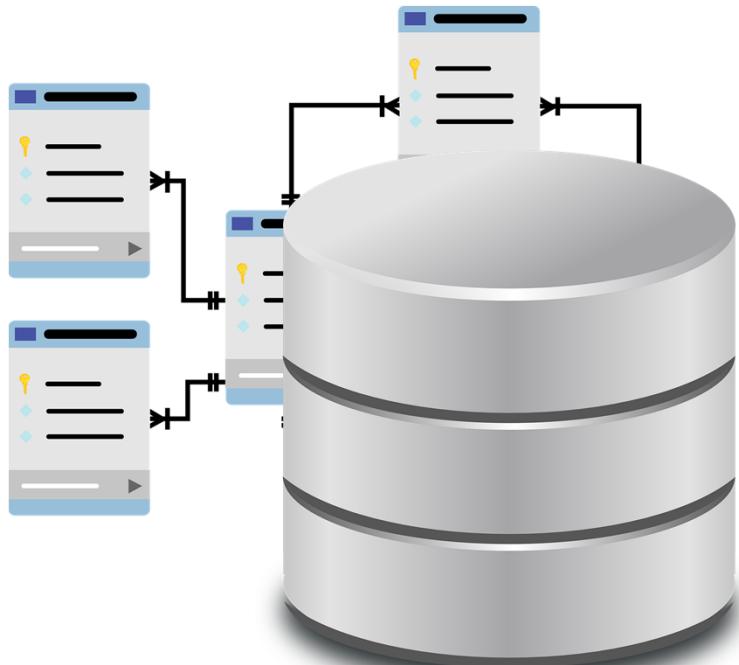
- Create Databases
- Database Queries
- Database Maintenance
- Provides Interface to Applications

Introduction to Data Science: Database Management System (DBMS)

Benefits

- Databases Integration
- Reduces Redundancy
- Enables Data Sharing
- Manages Integrity

Introduction to Data Science: Structured Query Language

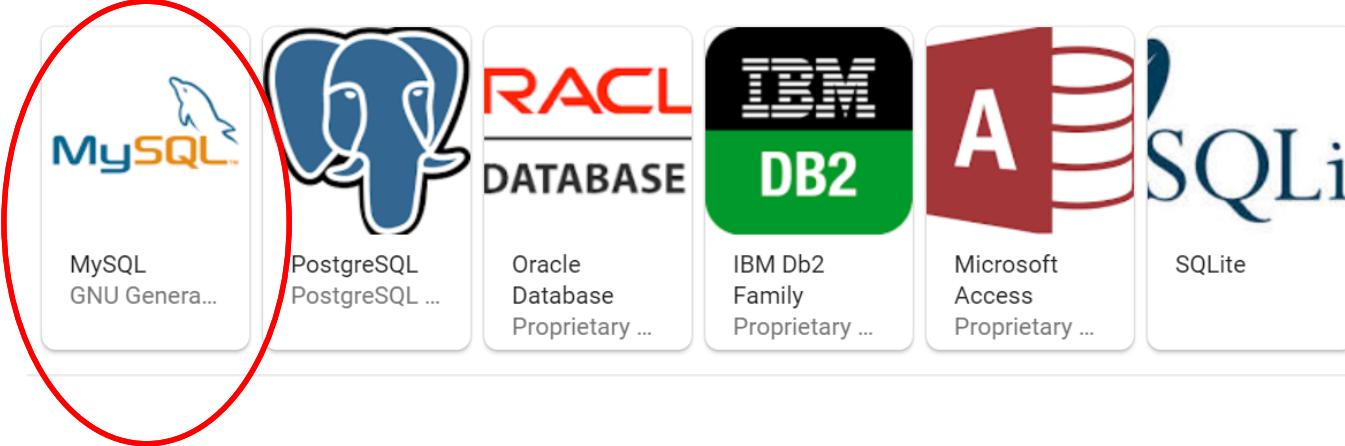


Structured Query Language (SQL)

- Developed by IBM (SEQUEL)
- Standard database programming language for interface with relational databases.
 - create, maintain, retrieve

Introduction to Data Science: SQL Database Management Systems

Database management system Software



What do they have in common?

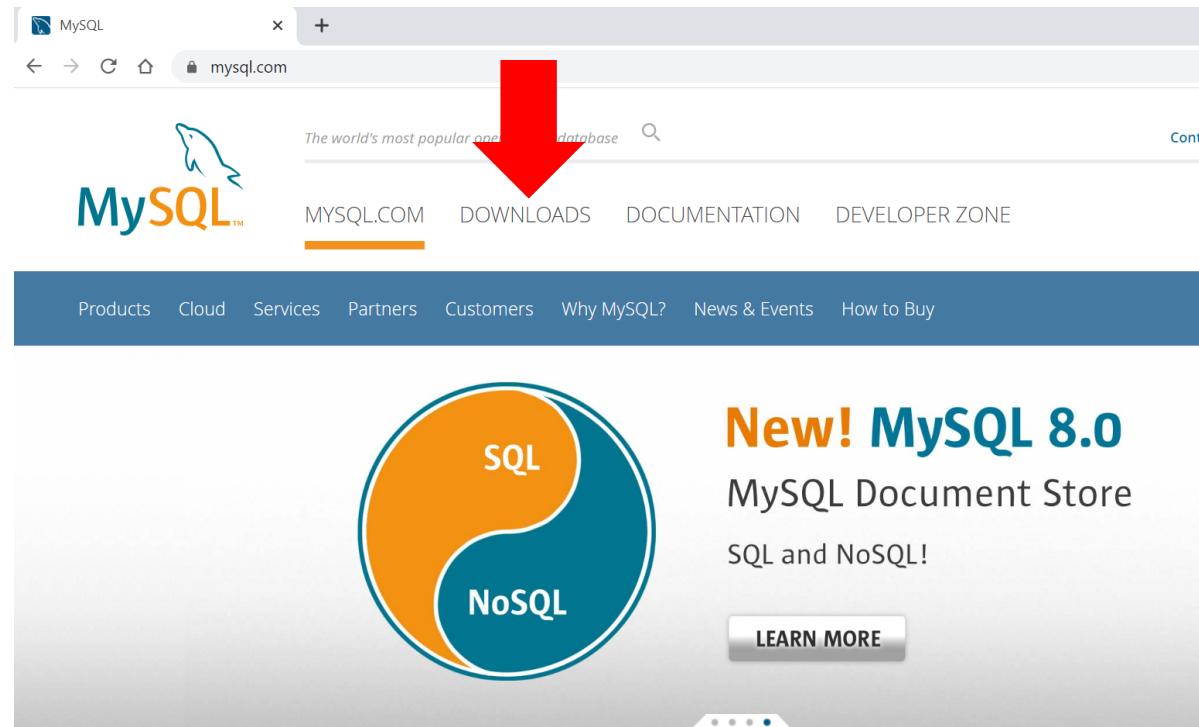
- Structured Query Language (SQL) as the database programming language

Introduction to Data Science: Why MySQL?



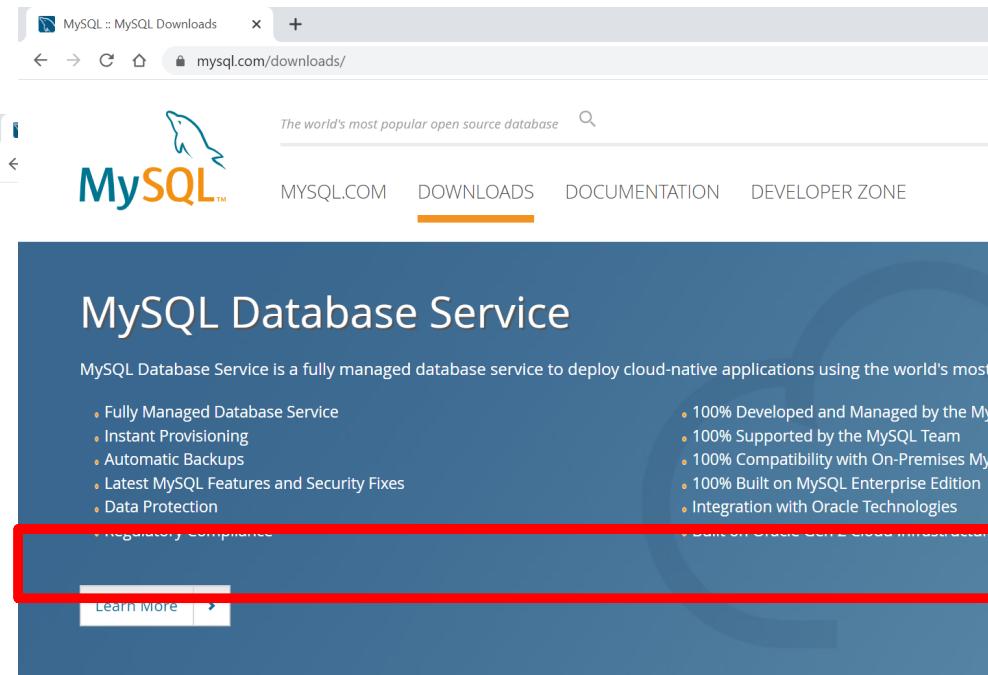
Rank			DBMS	Database Model
May 2020	Apr 2020	May 2019		
1.	1.	1.	Oracle 	Relational, Multi-model 
2.	2.	2.	MySQL 	Relational, Multi-model 
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 
4.	4.	4.	PostgreSQL 	Relational, Multi-model 
5.	5.	5.	MongoDB 	Document, Multi-model 
6.	6.	6.	IBM Db2 	Relational, Multi-model 
7.	7.	7.	Elasticsearch 	Search engine, Multi-model 
8.	8.	8.	Redis 	Key-value, Multi-model 
9.	9.	↑ 11.	SQLite 	Relational
10.	10.	↓ 9.	Microsoft Access	Relational

Introduction to Data Science: MySQL installation



URL: mysql.com

Introduction to Data Science: MySQL installation



The screenshot shows a web browser window displaying the MySQL Downloads page at mysql.com/downloads/. The page features the MySQL logo and navigation links for MySQL.COM, DOWNLOADS (which is highlighted with an orange bar), DOCUMENTATION, and DEVELOPER ZONE. The main content area is titled "MySQL Database Service" and describes it as a fully managed database service. It lists several benefits: Fully Managed Database Service, Instant Provisioning, Automatic Backups, Latest MySQL Features and Security Fixes, Data Protection, 100% Developed and Managed by the MySQL Team, 100% Supported by the MySQL Team, 100% Compatibility with On-Premises MySQL, 100% Built on MySQL Enterprise Edition, and Integration with Oracle Technologies. A red box highlights the "MySQL Database Service" section. At the bottom, there is a "Learn More" button.

[MySQL Community \(GPL\) Downloads »](#)

Introduction to Data Science: MySQL installation

A screenshot of a web browser window. The title bar says "MySQL :: MySQL Community Downloads". The address bar shows "dev.mysql.com/downloads/". Below the address bar are standard navigation icons: back, forward, refresh, and home. The main content area displays a list of MySQL download options.

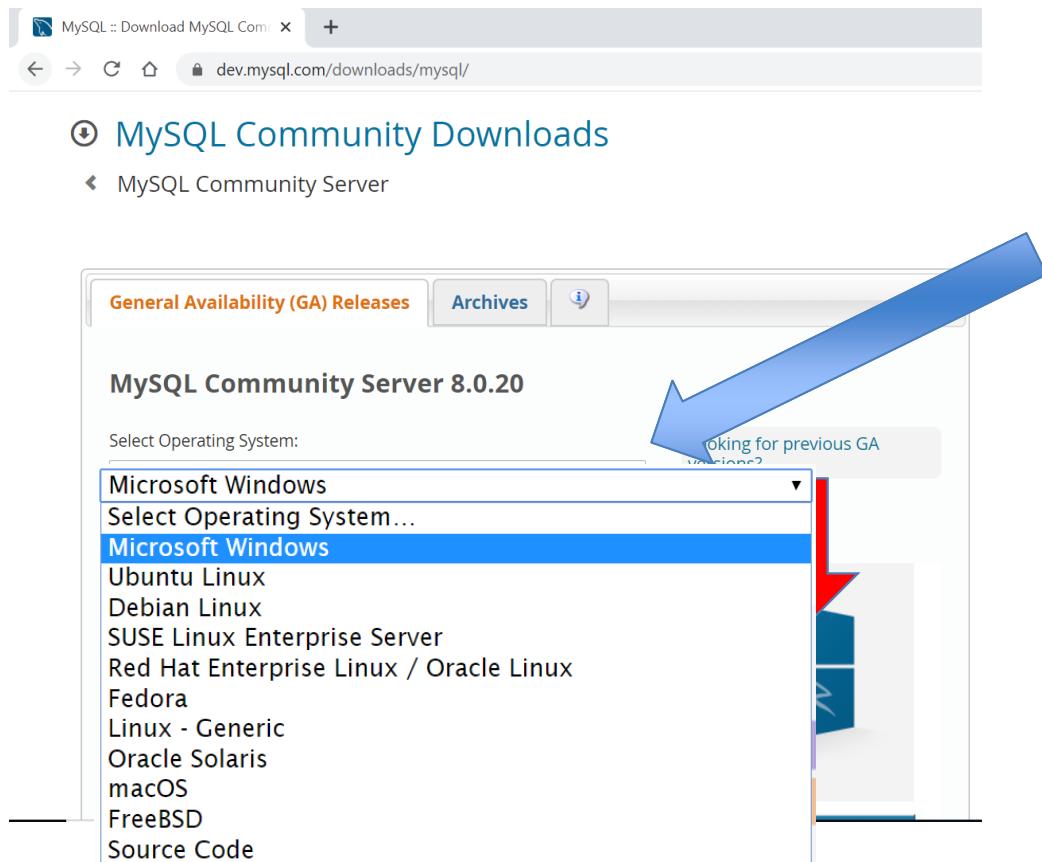
MySQL Community Downloads

- [MySQL Yum Repository](#)
- [MySQL APT Repository](#)
- [MySQL SUSE Repository](#)
- [MySQL Community Server](#)
- [MySQL Cluster](#)
- [MySQL Router](#)
- [MySQL Shell](#)
- [MySQL Workbench](#)
- [MySQL Installer for Windows](#)
- [MySQL for Excel](#)
- [MySQL for Visual Studio](#)
- [MySQL Notifier](#)
- [C API \(libmysqlclient\)](#)
- [Connector/C++](#)
- [Connector/J](#)
- [Connector/.NET](#)
- [Connector/Node.js](#)
- [Connector/ODBC](#)
- [Connector/Python](#)
- [MySQL Native Driver for PHP](#)
- [MySQL Benchmark Tool](#)
- [Time zone description tables](#)
- [Download Archives](#)



© 2020, Oracle Corporation and/or its affiliates

Introduction to Data Science: MySQL installation



Introduction to Data Science: MySQL installation

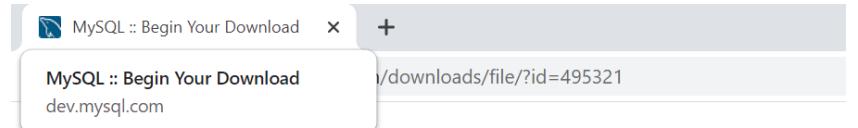
The screenshot shows a web browser window displaying the MySQL Community Downloads page. The URL in the address bar is dev.mysql.com/downloads/windows/installer/8.0.html. The page title is "MySQL :: Download MySQL Instal".

The main content area is titled "MySQL Installer 8.0.20". It features a "Select Operating System:" dropdown menu set to "Microsoft Windows". Below the dropdown, there are two download options:

- Windows (x86, 32-bit), MSI Installer**: Version 8.0.20, file size 24.4M. The "Download" button is highlighted with a red box.
- Windows (x86, 32-bit), MSI Installer**: Version 8.0.20, file size 420.6M. The "Download" button is visible but not highlighted.

A note at the bottom left of the page says: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

Introduction to Data Science: MySQL installation



MySQL Community Downloads

Login Now or Sign Up for a free account.

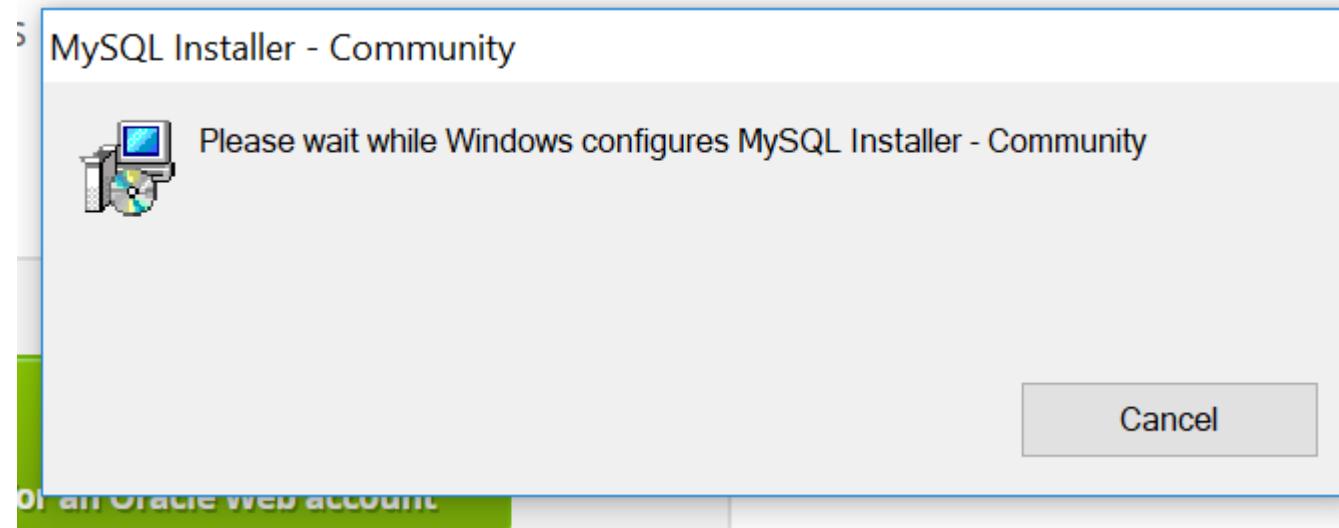
An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

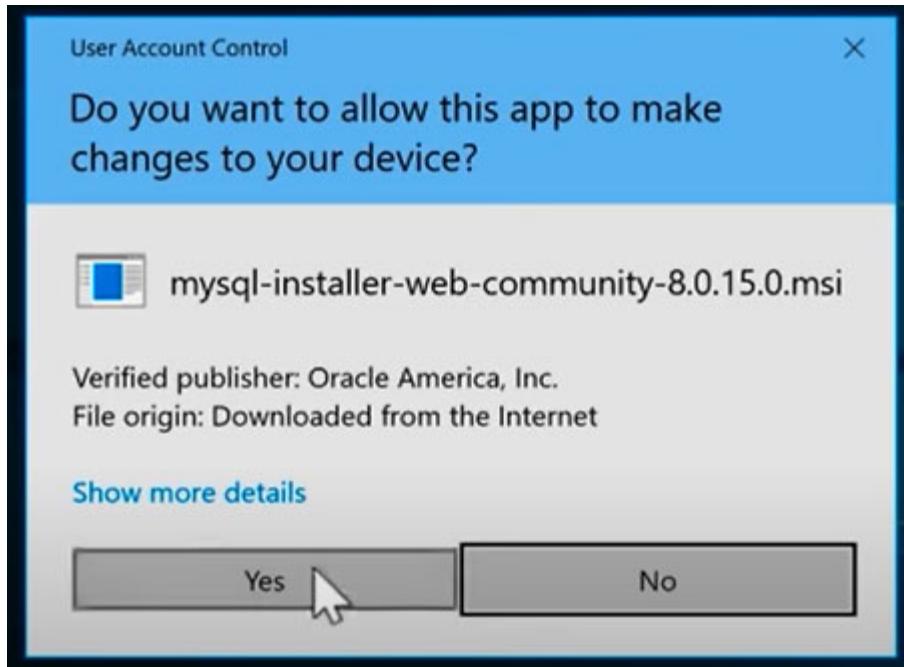
A screenshot of a login interface. It features two prominent buttons: a blue "Login »" button with the subtext "using my Oracle Web account" and a green "Sign Up »" button with the subtext "for an Oracle Web account". Below these buttons, there is explanatory text: "MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions." A small computer icon is visible in the top left corner of the interface.



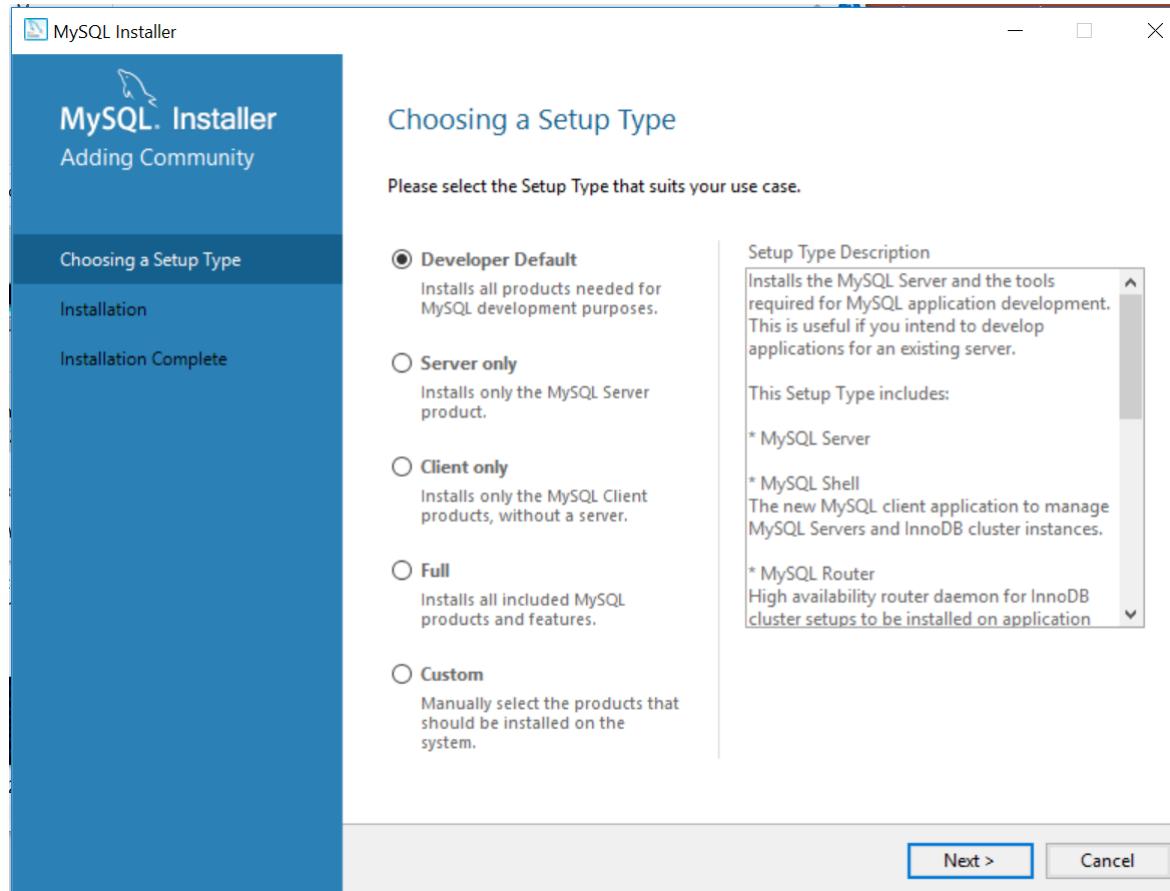
Introduction to Data Science: MySQL installation



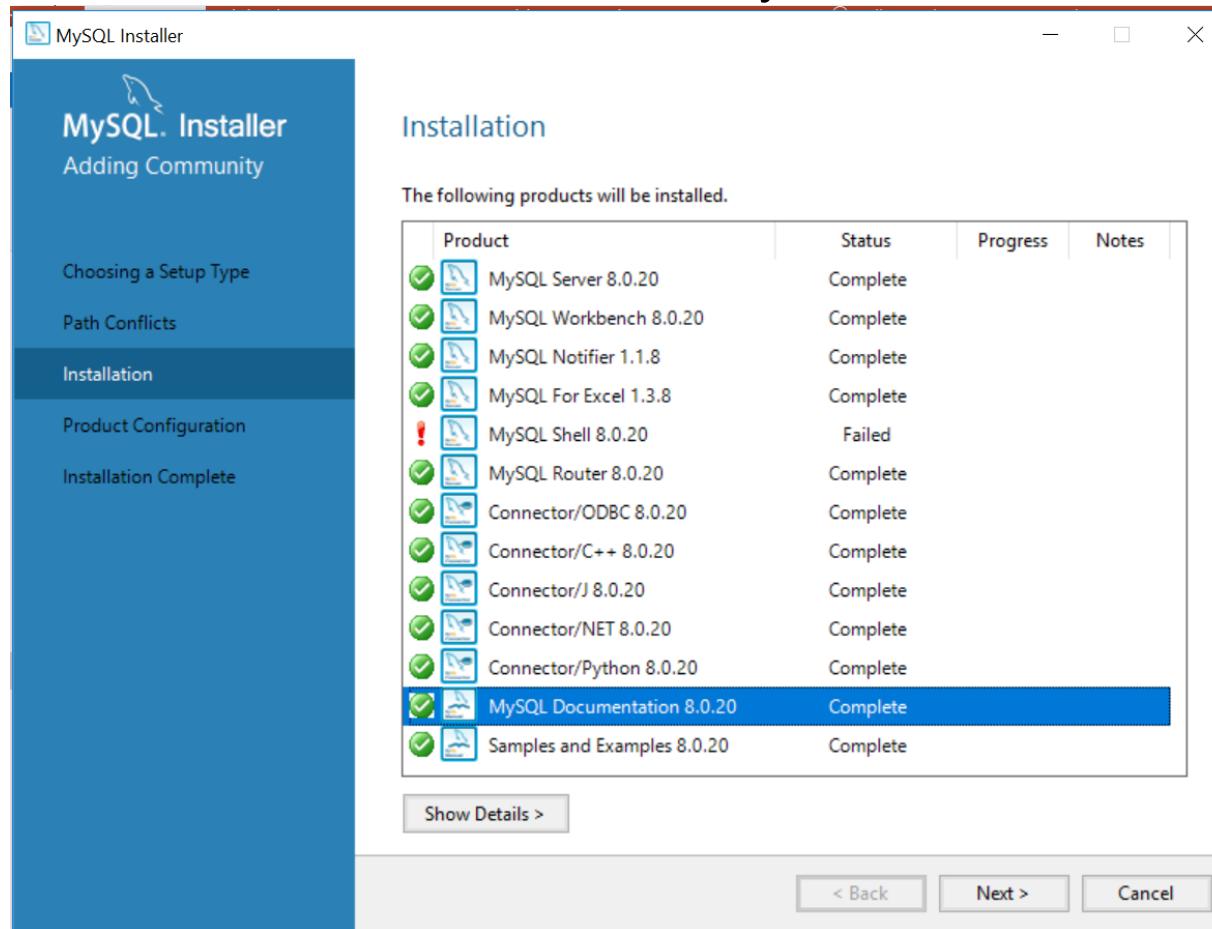
Introduction to Data Science: MySQL installation



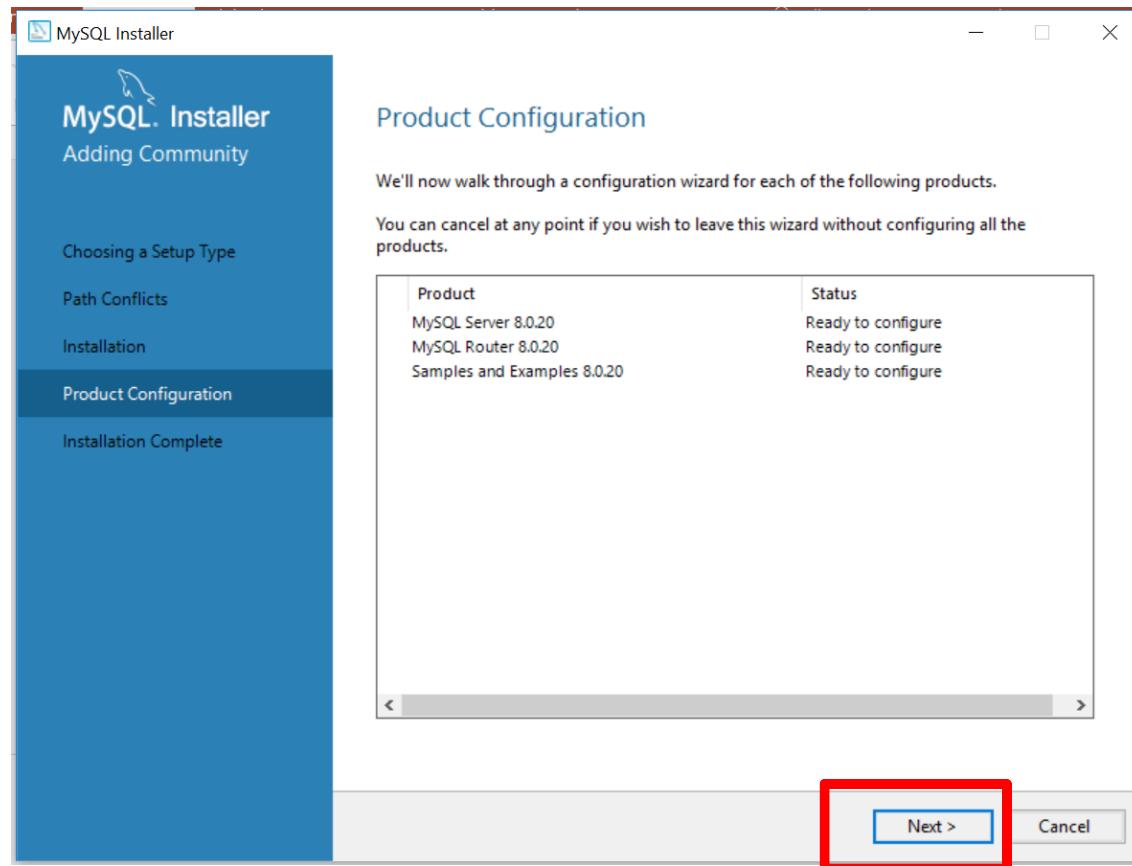
Introduction to Data Science: MySQL installation



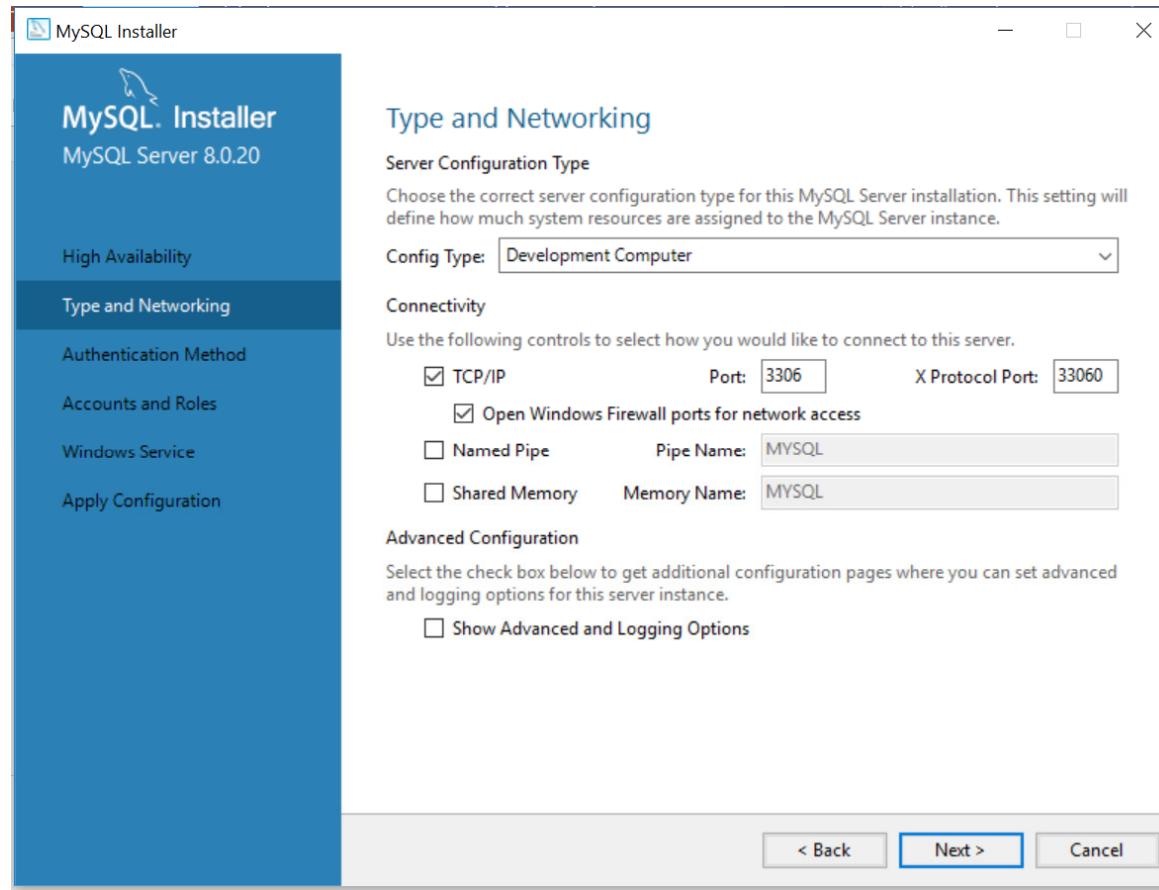
Introduction to Data Science: MySQL installation



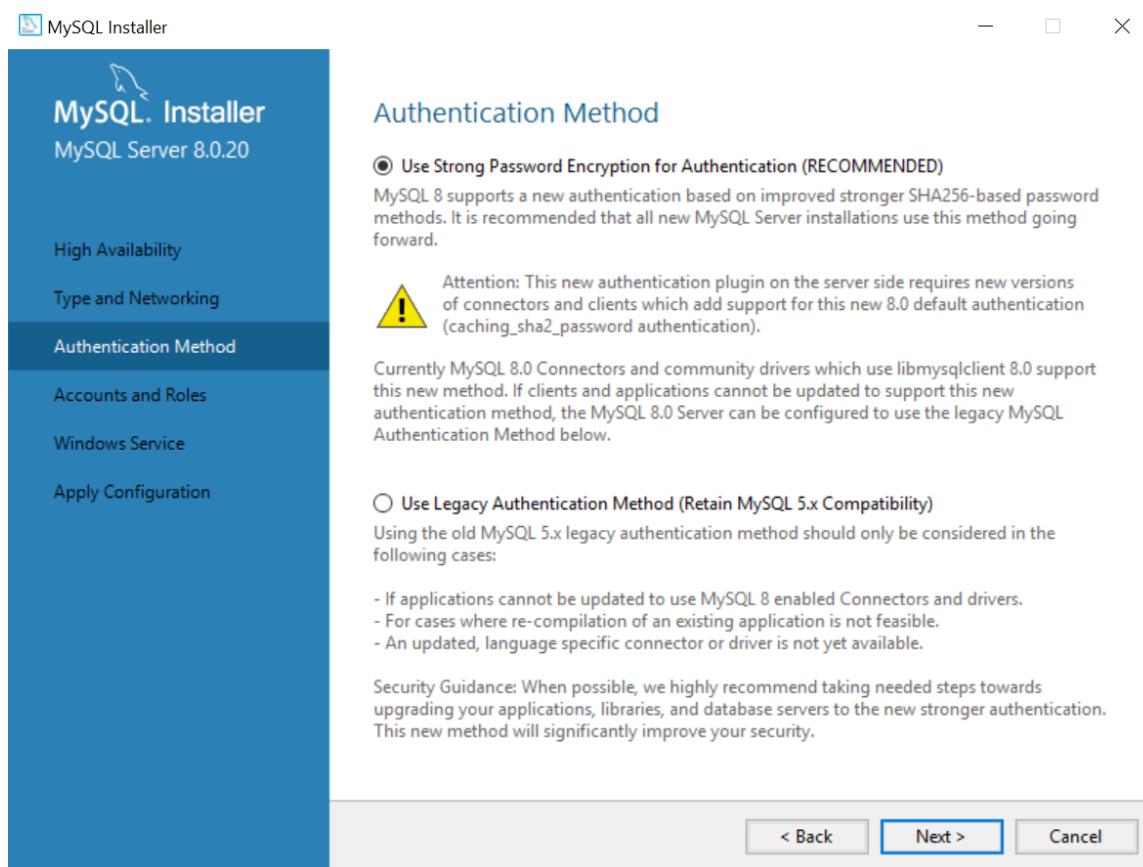
Introduction to Data Science: MySQL installation



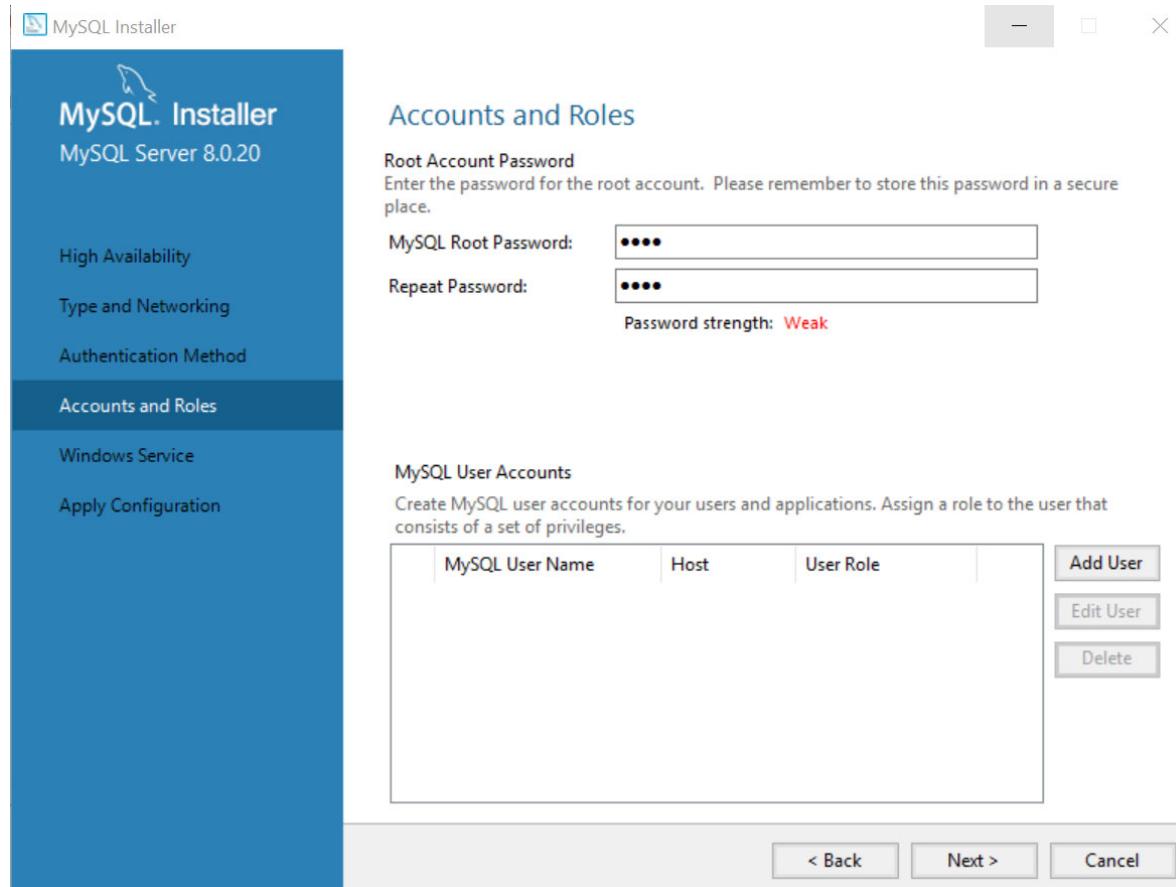
Introduction to Data Science: MySQL installation



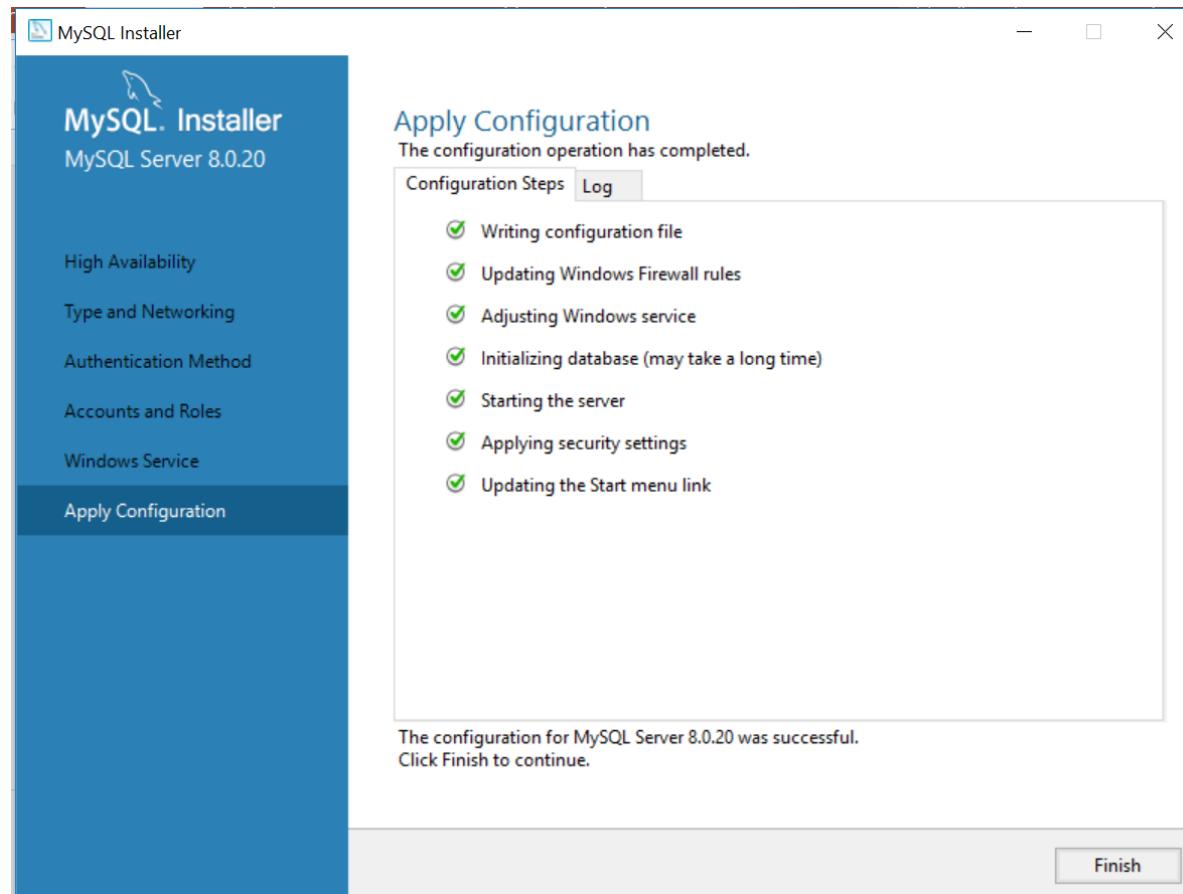
Introduction to Data Science: MySQL installation



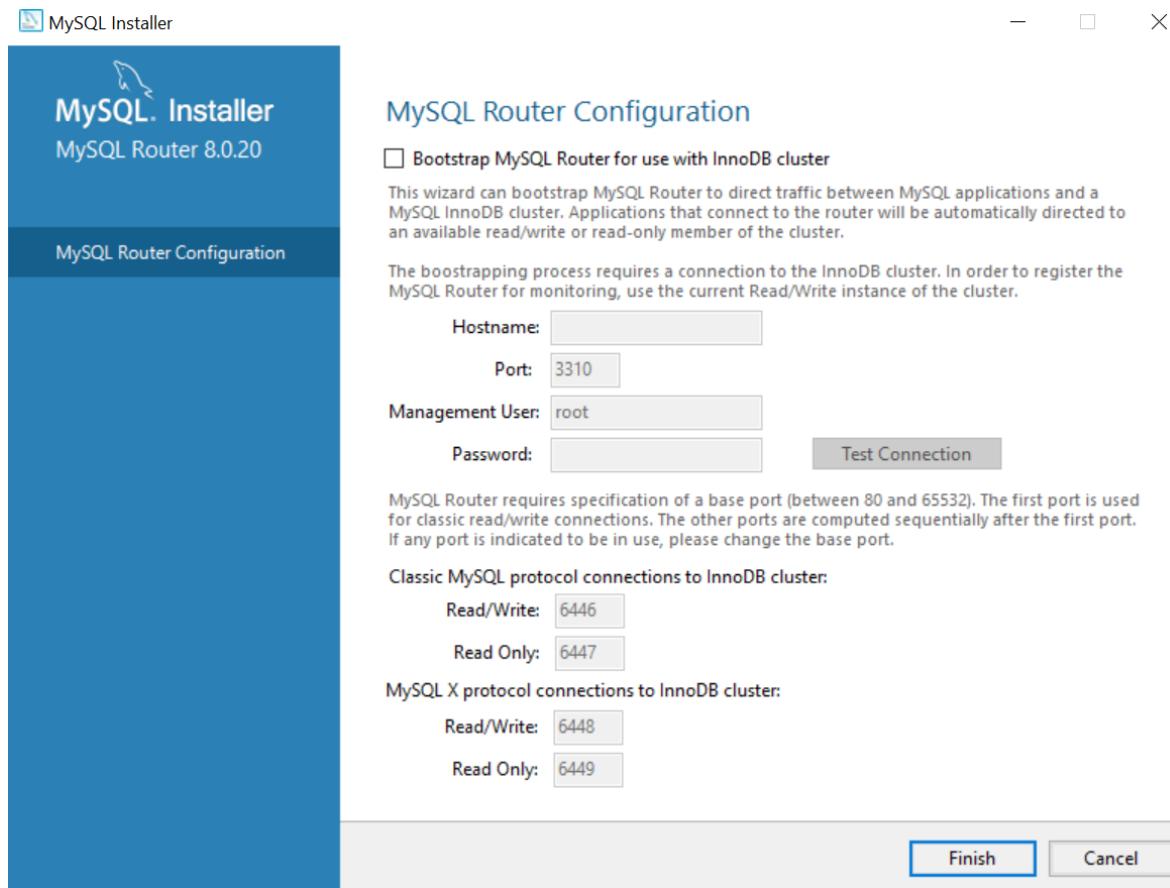
Introduction to Data Science: MySQL installation



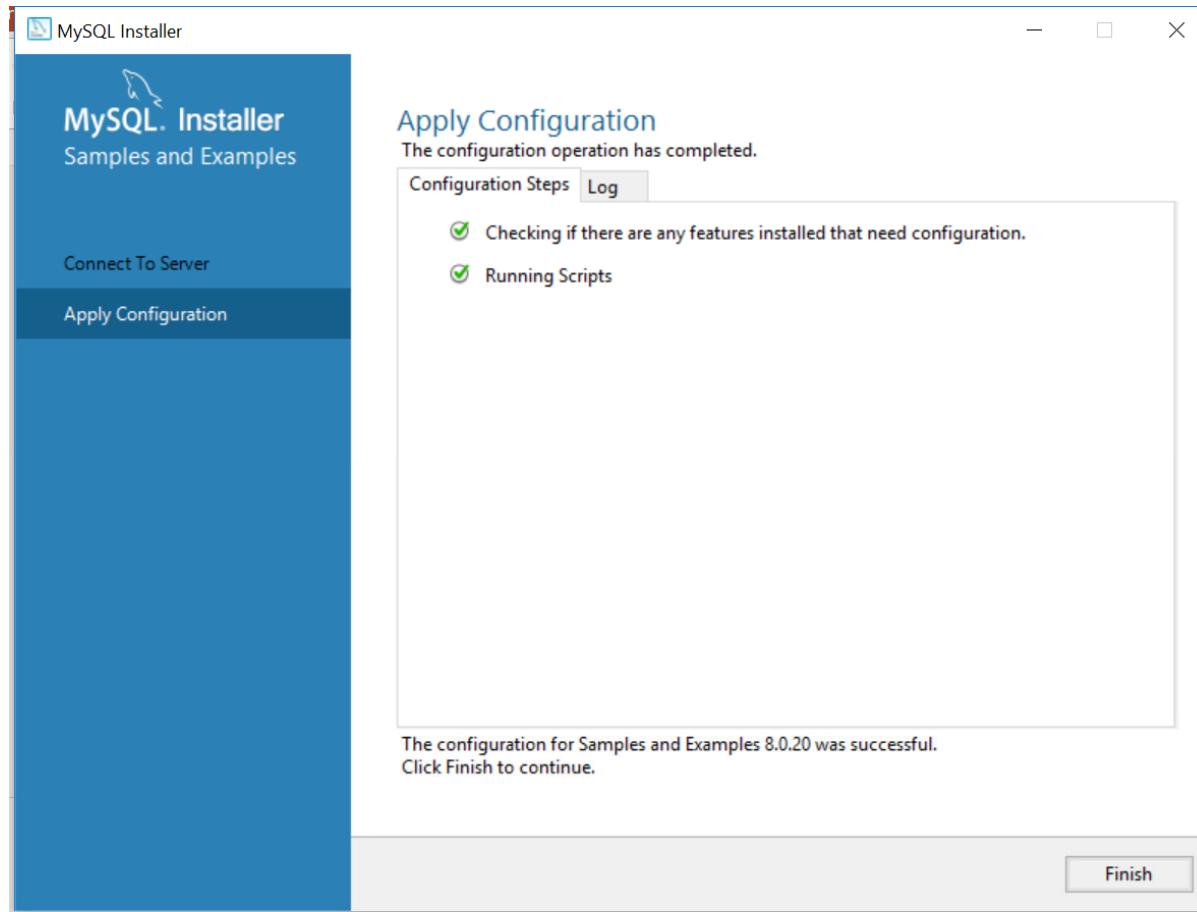
Introduction to Data Science: MySQL installation



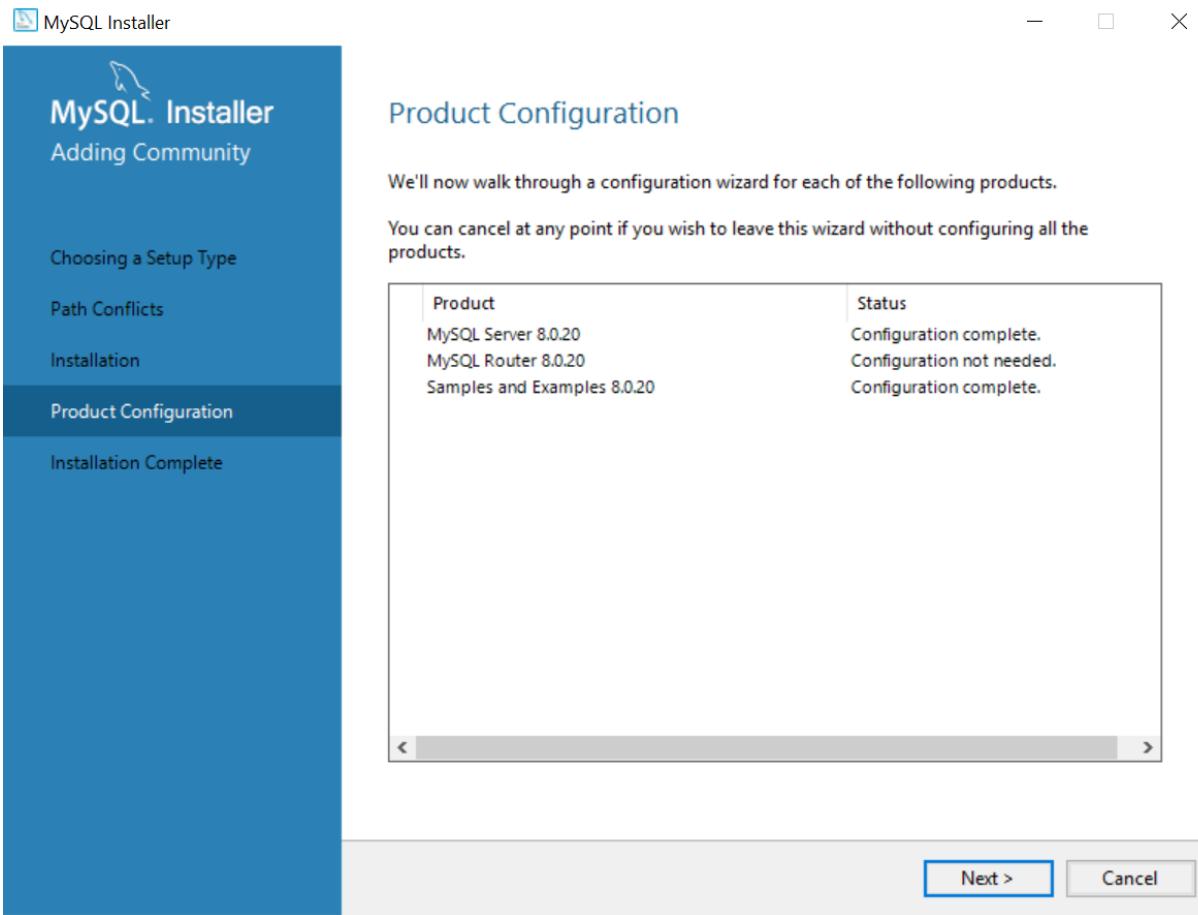
Introduction to Data Science: MySQL installation



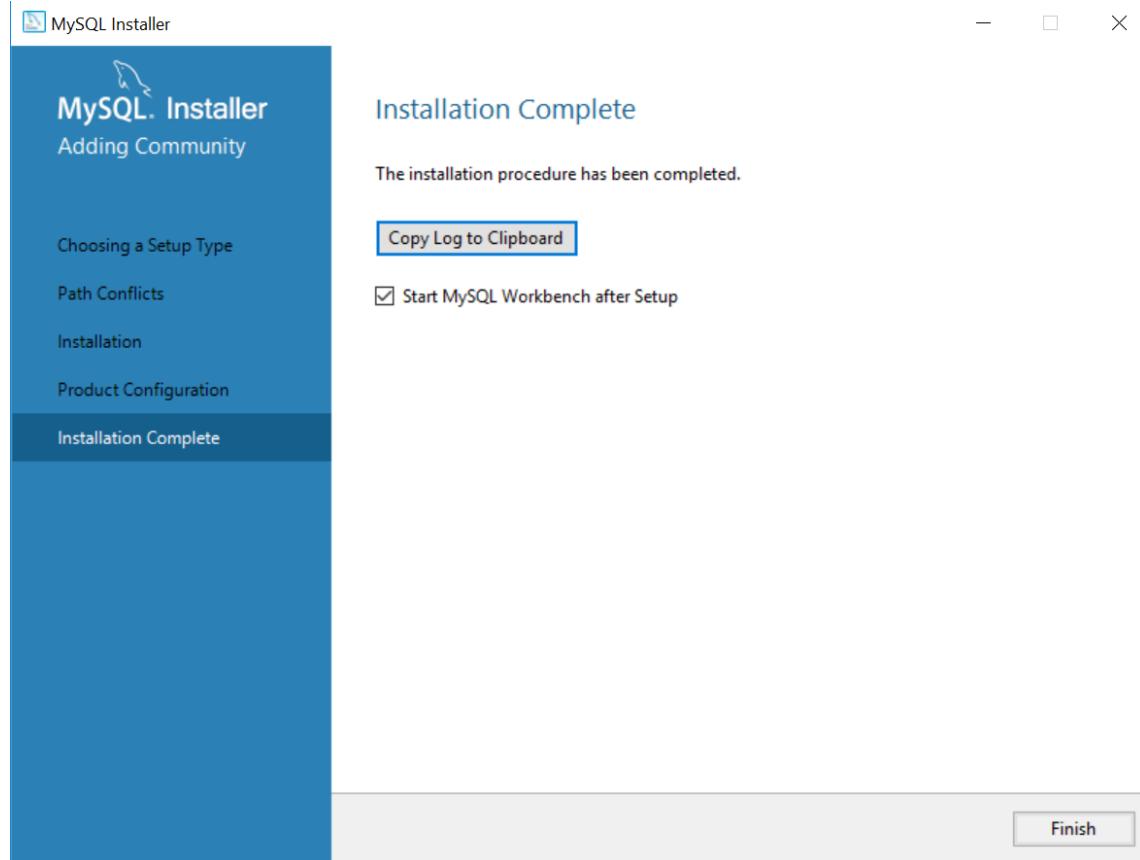
Introduction to Data Science: MySQL installation



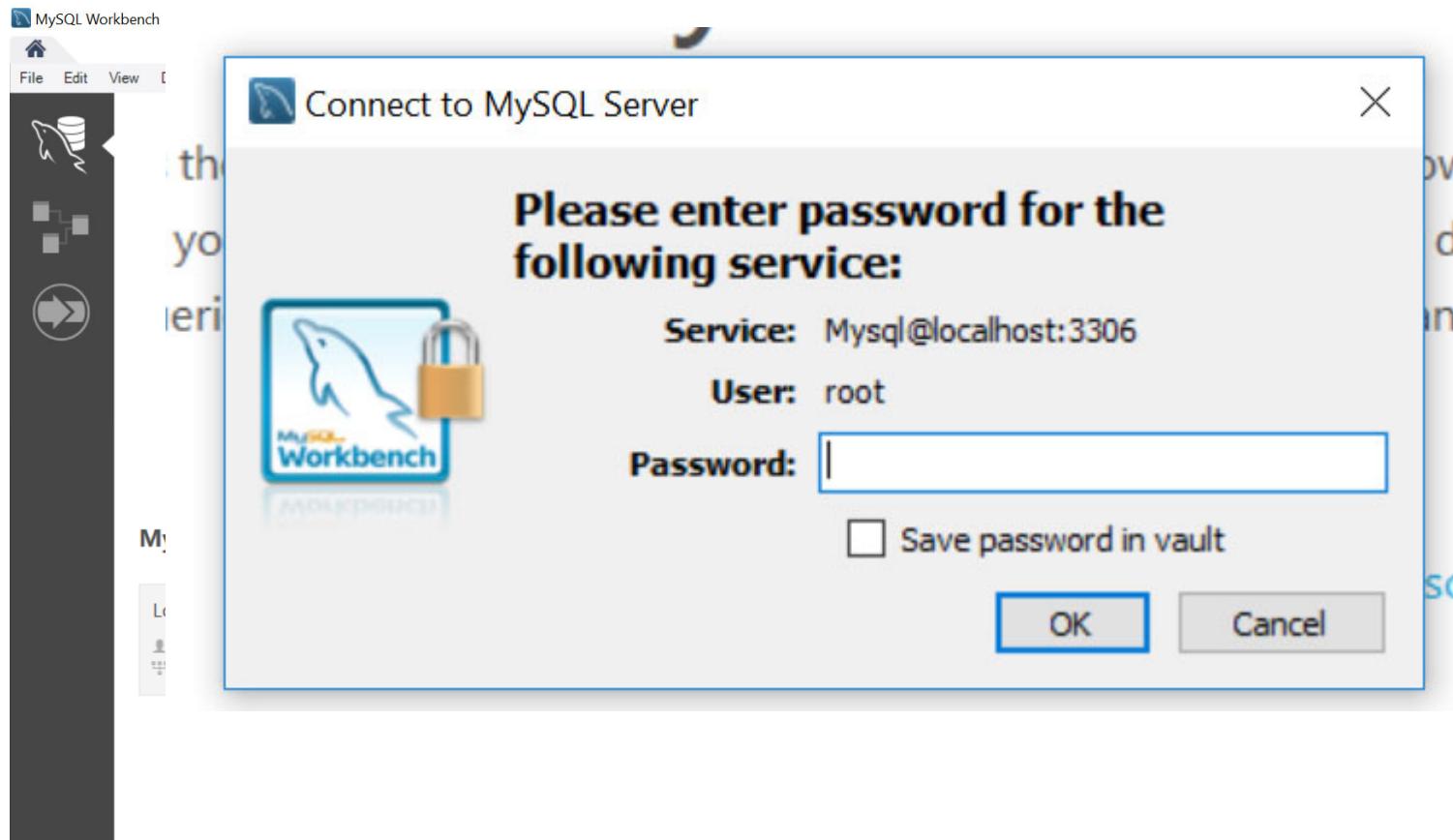
Introduction to Data Science: MySQL installation



Introduction to Data Science: MySQL installation



Introduction to Data Science: MySQL installation



MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

No object selected

Object Info Session

Query 1 ×

SQLAdditions

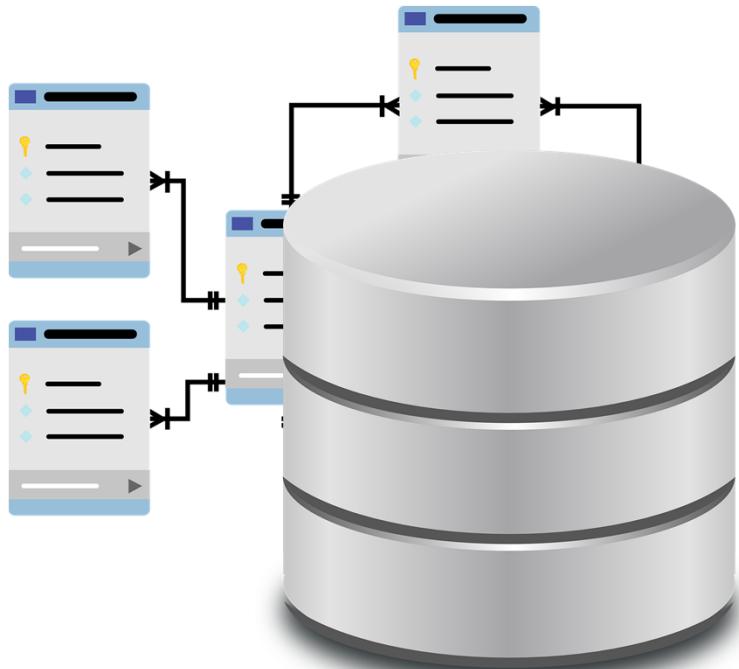
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Action Output

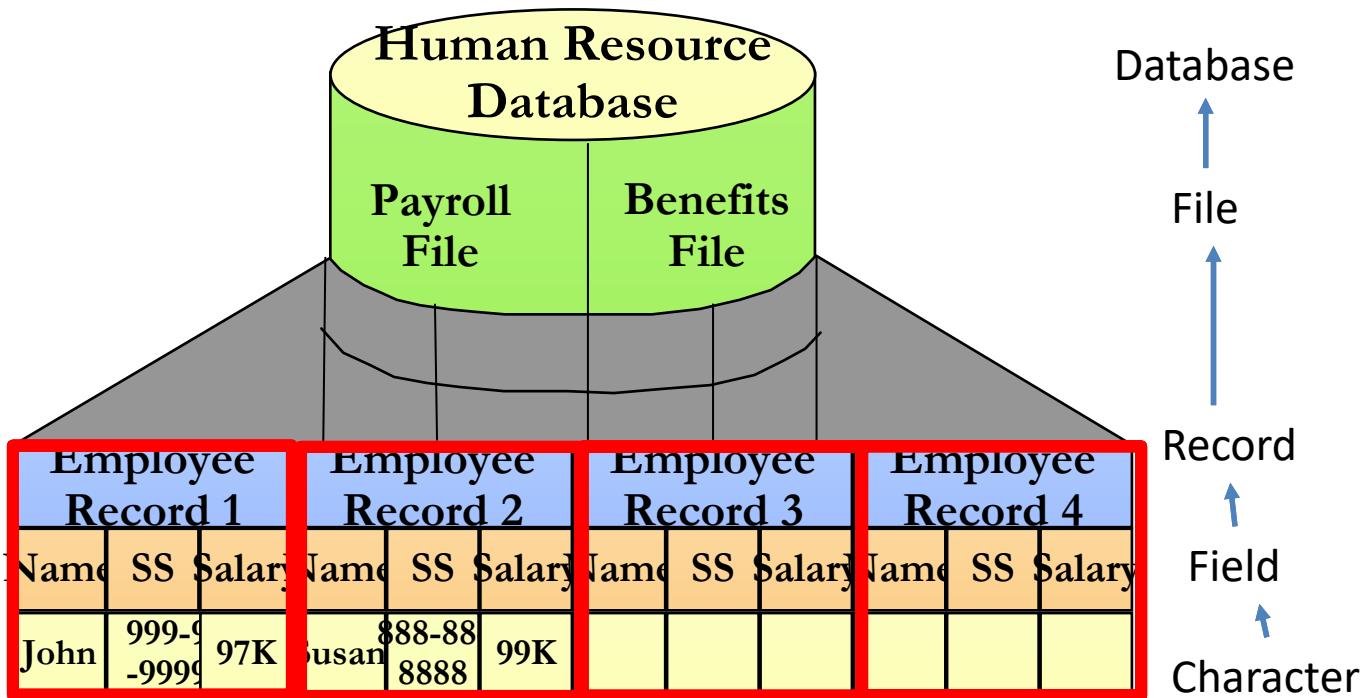
#	Time	Action	Message	Duration / Fetch

Introduction to Data Science: Structured Query Language

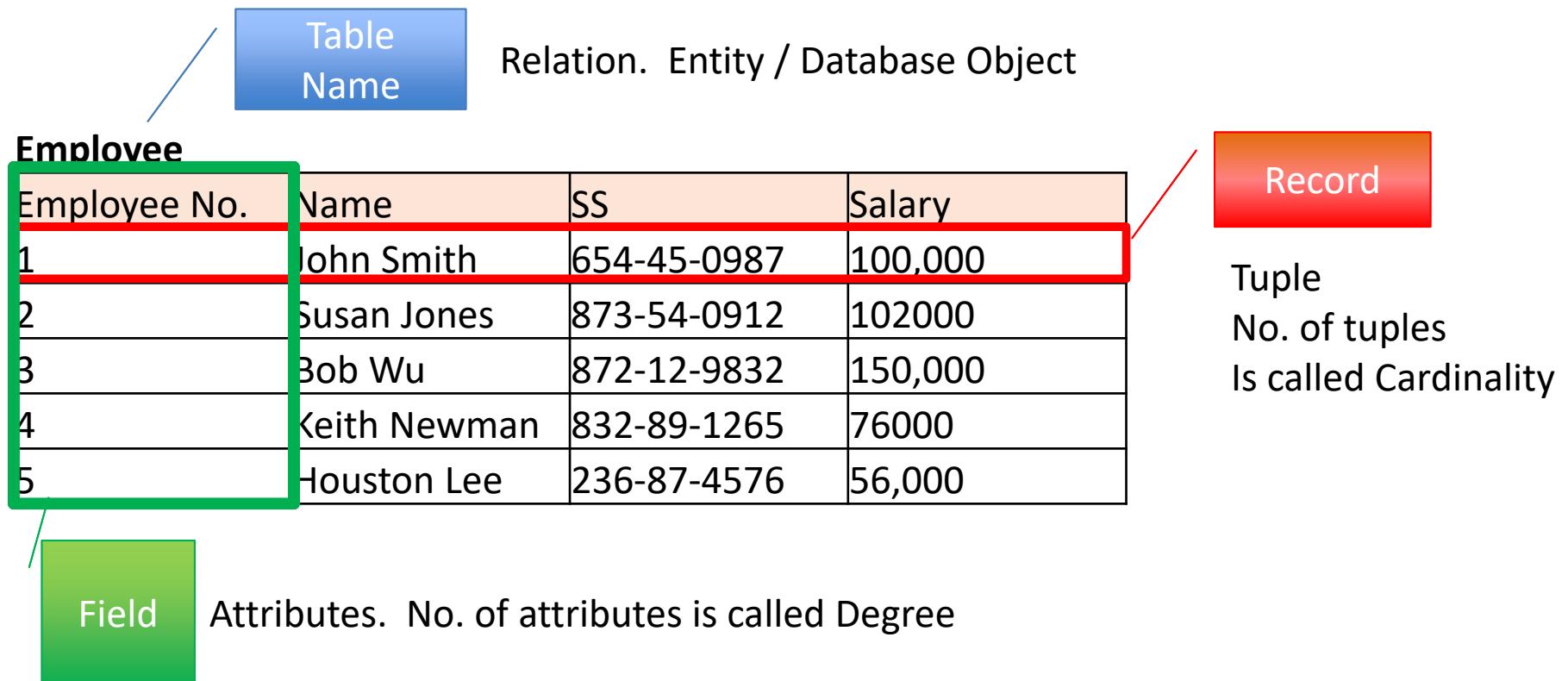


Structured Query Language (SQL)

Introduction to Data Science: SQL



Introduction to Data Science: SQL



Introduction to Data Science: SQL

Employee

Employee No.	Name	SS	Salary
1	John Smith	654-45-0987	100,000
2	Susan Jones	873-54-0912	102000
3	Bob Wu	872-12-9832	150,000
4	Keith Newman	832-89-1265	76000
5	Houston Lee	236-87-4576	56,000

Primary Key

Surrogate key

Introduction to Data Science: SQL

Employee

Employee No.	Name	Salary
654-45-0987	John Smith	100,000
873-54-0912	Susan Jones	102000
872-12-9832	Bob Wu	150,000
832-89-1265	Keith Newman	76000
236-87-4576	Houston Lee	56,000

Primary Key

Natural key

Introduction to Data Science: SQL

Employee

Employee No.	Name	Salary	Division ID
654-45-0987	John Smith	100,000	10
873-54-0912	Susan Jones	102000	20
872-12-9832	Bob Wu	150,000	30
832-89-1265	Keith Newman	76000	20
236-87-4576	Houston Lee	56,000	40

Manager

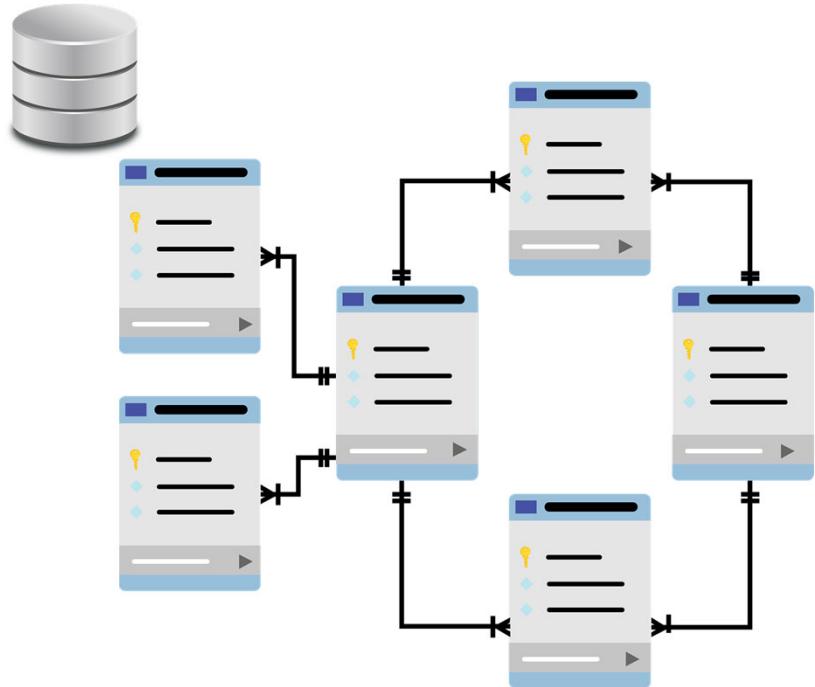
Manager ID	Name	Division ID
100	John Smith	10
200	Susan Jones	20
300	Bob Wu	20
400	Keith Newman	10
500	Houston Lee	50

Division

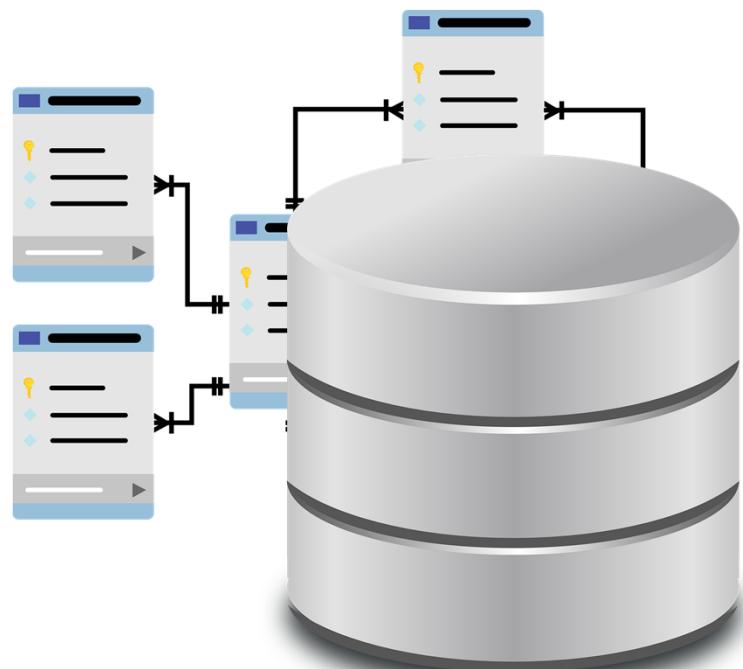
Division ID	Division Name	Budget
10	Sales	5,000,000
20	Marketing	1000000000
30	Product	10,000,000,000
40	Admin	500000
50	Legal	200,000

Foreign Key

Introduction to Data Science: Structured Query Language



Introduction to Data Science: SQL DDL



Data Definition Language DDL

Introduction to Data Science: SQL DDL

- Define Database Structure
 - Create databases, tables
 - Add columns
 - Delete

Introduction to Data Science: SQL DDL

To Create a Database:

```
CREATE DATABASE myschool;
```

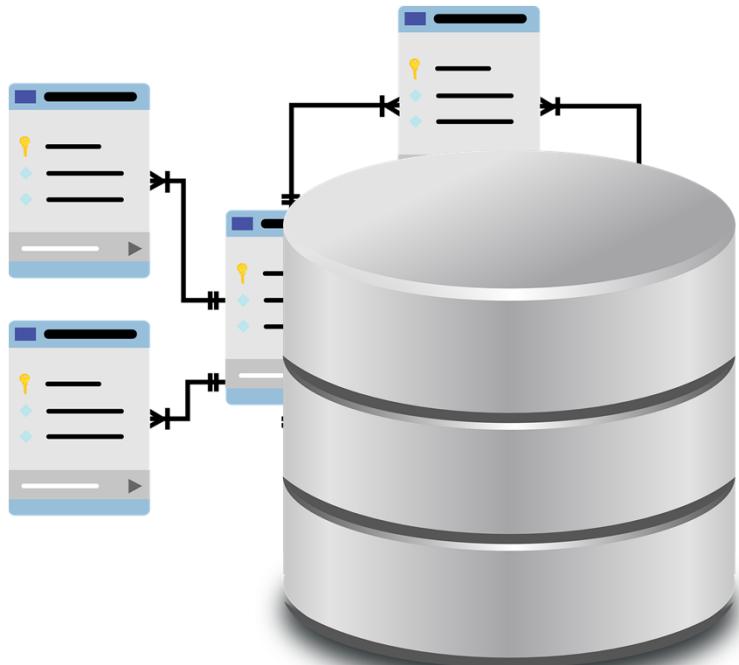
```
CREATE SCHEMA myschool;
```

Introduction to Data Science: SQL DDL

To Create a Table:

```
CREATE TABLE `myschool`.`employee` (  
  `id` INT NOT NULL, `name` VARCHAR(45)  
  NOT NULL, `salary` VARCHAR(45) NOT  
  NULL, PRIMARY KEY (`id`));
```

Introduction to Data Science: SQL Data Definition Language



Syntax and Pitfalls

Introduction to Data Science: SQL Create and Use Database

Let's create a database named "mystore"

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL

-- Create database for mystore

`CREATE SCHEMA mystore;`

-- Create table named inventory in mystore

`CREATE TABLE mystore.inventory(Item_no INT NOT NULL, Item_name VARCHAR(100) NOT NULL, Unit_Price INT NOT NULL, Inventory INT, PRIMARY KEY (Item_no));`

-- Populate table with values/data

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (2321, 'Dell Laptop', 1500, 56);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (5432, 'Seagate Drive', 200, 100);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (5674, 'Kingston USB Drive', 70, 500);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (8542, 'Backpack', 100, 45);`

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL DDL

-- Select database to use

USE mystore;

/* once USE mystore is executed, we can eliminate
the dot operator and database name */

```
CREATE TABLE inventory(Item_no INT NOT NULL,  
Item_name VARCHAR(100) NOT NULL, Unit_Price INT  
NOT NULL, Inventory INT, PRIMARY KEY (Item_no));
```

-- Insert new product with null value

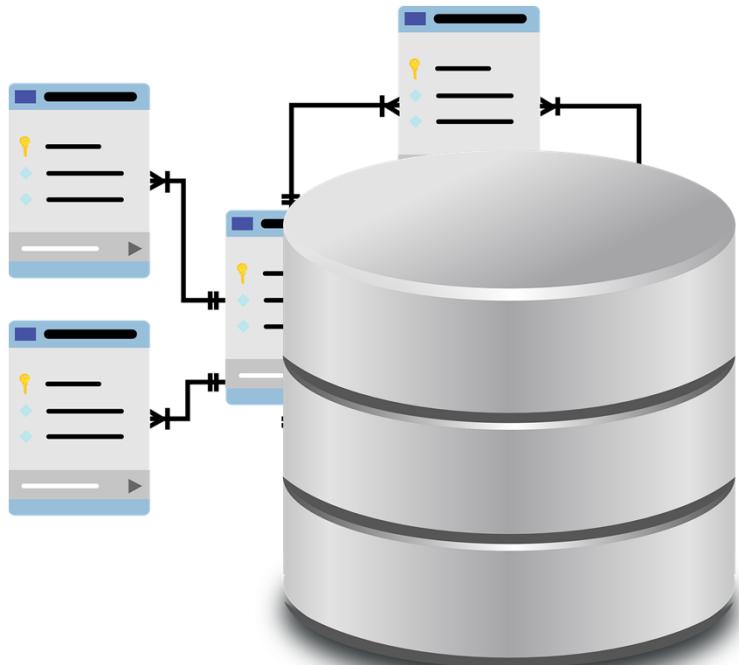
```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price, Inventory) VALUES (2348, 'HP Laptop', 1000,  
null);
```

-- Insert new product with no null columns only

```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price) VALUES (7344, 'Lenovo Laptop', 988);
```

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL Data Definition Language



Syntax and Pitfalls

Introduction to Data Science: SQL DDL

-- Select database to use

USE mystore;

/* once USE mystore is executed, we can eliminate
the dot operator and database name */

```
CREATE TABLE inventory(Item_no INT NOT NULL,  
Item_name VARCHAR(100) NOT NULL, Unit_Price INT  
NOT NULL, Inventory INT, PRIMARY KEY (Item_no));
```

-- Insert new product with null value

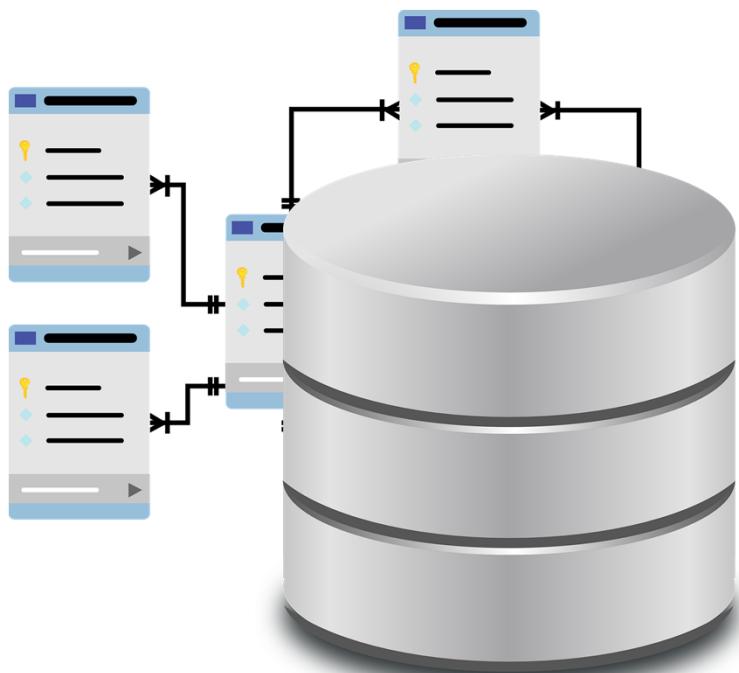
```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price, Inventory) VALUES (2348, 'HP Laptop', 1000,  
null);
```

-- Insert new product with no null columns only

```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price) VALUES (7344, 'Lenovo Laptop', 988);
```

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL Build Query Script



Build Query Script on MySQL Workbench

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL

-- Create database for mystore

`CREATE SCHEMA mystore;`

-- Create table named inventory in mystore

`CREATE TABLE mystore.inventory(Item_no INT NOT NULL, Item_name VARCHAR(100) NOT NULL, Unit_Price INT NOT NULL, Inventory INT, PRIMARY KEY (Item_no));`

-- Populate table with values/data

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (2321, 'Dell Laptop', 1500, 56);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (5432, 'Seagate Drive', 200, 100);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (5674, 'Kingston USB Drive', 70, 500);`

`INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) VALUES (8542, 'Backpack', 100, 45);`

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Introduction to Data Science: SQL DDL

-- Select database to use

USE mystore;

/* once USE mystore is executed, we can eliminate
the dot operator and database name */

```
CREATE TABLE inventory(Item_no INT NOT NULL,  
Item_name VARCHAR(100) NOT NULL, Unit_Price INT  
NOT NULL, Inventory INT, PRIMARY KEY (Item_no));
```

-- Insert new product with null value

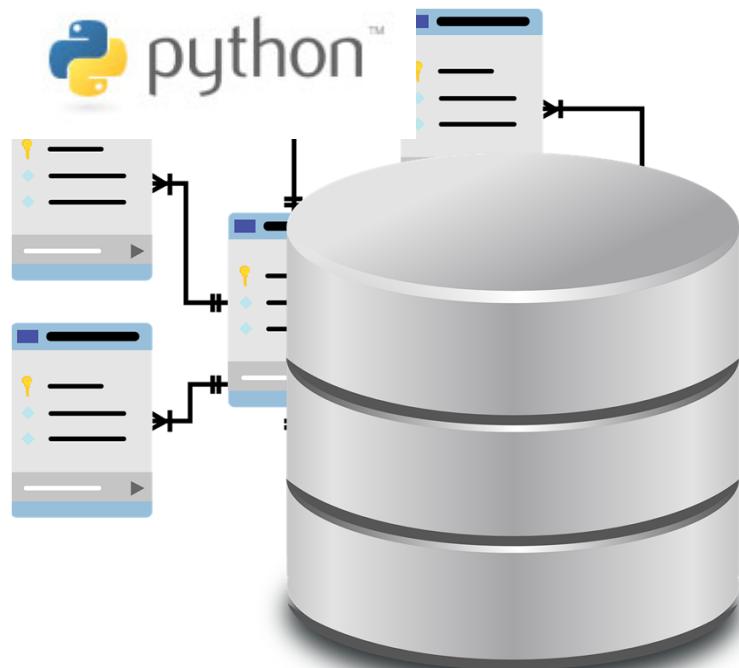
```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price, Inventory) VALUES (2348, 'HP Laptop', 1000,  
null);
```

-- Insert new product with no null columns only

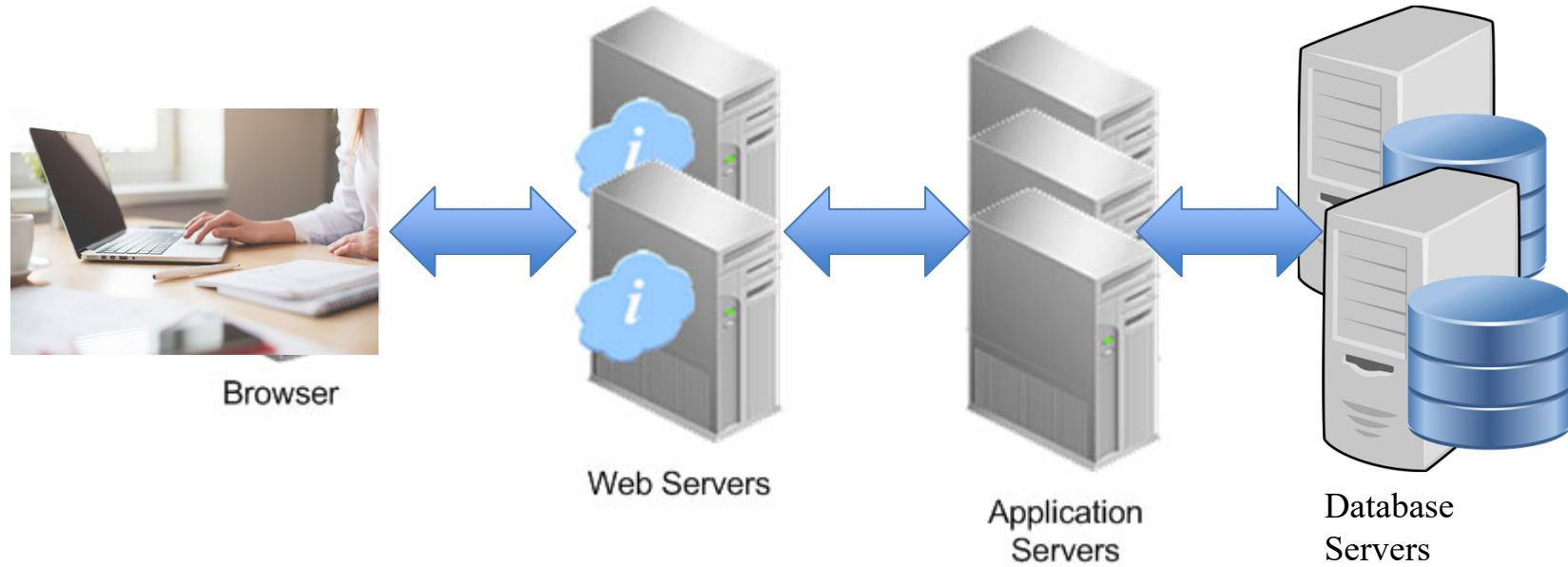
```
INSERT INTO inventory (Item_no, Item_name,  
Unit_Price) VALUES (7344, 'Lenovo Laptop', 988);
```

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

Data Science: Python and MySQL Integration



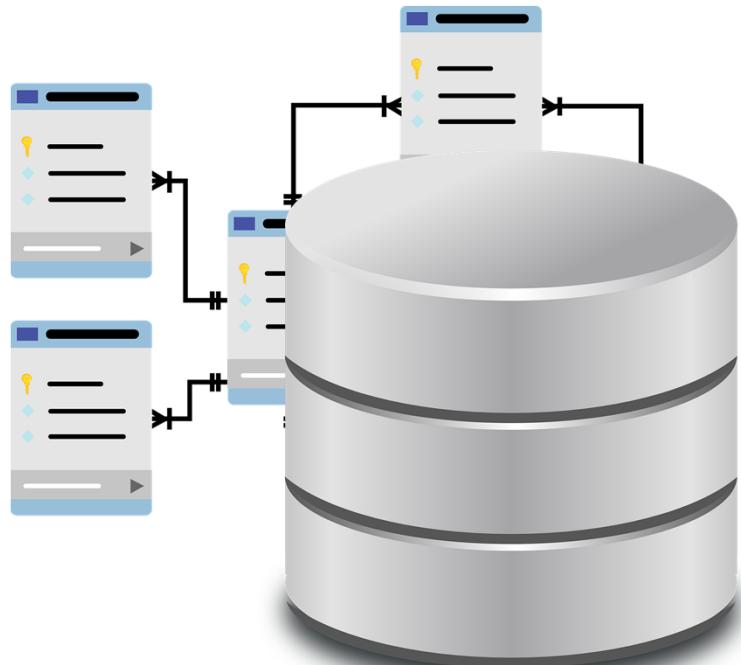
Data Science: Python and MySQL Integration



Data Science: Python and MySQL Integration

```
: import mysql.connector as sq  
:  
: mydb=sq.connect(host="localhost",user="root",passwd="ucla", buffered=True)  
:  
: mycursor = mydb.cursor()  
:  
: mycursor.execute("SHOW DATABASES")  
:  
: for db in mycursor:  
:     print (db)
```

SQL Create and Insert using Executemany



Create and Insert executemany

Item_no	Item_name	Unit_Price	Inventory
2321	Dell Laptop	1500	56
5432	Seagate Drive	200	100
5674	Kingston USB Drive	70	500
8542	Backpack	100	45

```
import mysql.connector as sq

mydb=sq.connect(host="localhost",user="root",passwd="ucla", buffered=True)

mycursor = mydb.cursor()

mycursor.execute("CREATE SCHEMA mystore")

mycursor.execute("CREATE TABLE mystore.inventory(Item_no INT NOT NULL, Item_name VARCHAR(100) NOT NULL, \
Unit_Price INT NOT NULL, Inventory INT, PRIMARY KEY (Item_no))")

mycursor.execute("INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) \
VALUES (2321, 'Dell Laptop', 1500, 56)")
mycursor.execute("INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) \
VALUES (5432, 'Seagate Drive', 200, 100)")
mycursor.execute("INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) \
VALUES (5674, 'Kingston USB Drive', 70, 500)")
mycursor.execute("INSERT INTO mystore.inventory (Item_no, Item_name, Unit_Price, Inventory) \
VALUES (8542, 'Backpack', 100, 45)")

mydb.commit()
```

```
import mysql.connector as sq

mydb=sq.connect(host="localhost",user="root",passwd="ucla", buffered=True)

mycursor = mydb.cursor()

mycursor.execute("CREATE SCHEMA mystore")

mycursor.execute("CREATE TABLE mystore.inventory(Item_no INT NOT NULL, Item_name VARCHAR(100) NOT NULL, \
Unit_Price INT NOT NULL, Inventory INT, PRIMARY KEY (Item_no))")

SQLCMD = "INSERT INTO PyDB.PyTABLE (id, name, salary)VALUES(%s,%s,%s)"
values = [(100,'John Hall',12300.55),(101,'John Smith',10300.15),(102,'Susan Jones',11500.57)]

mycursor.executemany(SQLCMD, values)

mydb.commit()
```